

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer program product, tangibly embodied in an information carrier, comprising instructions operable to:
compile a procedural source code program to generate a compiled program having instructions to create a runtime data type having a compound structure of referenced data types and having instructions to use the runtime data type;
create, at compile time, data type definitions for the referenced data types included in the compiled program and known at compile time;
execute, at runtime, the instructions to create a runtime data type having a compound structure by creating a runtime data type definition ~~from based on~~ the compound structure of referenced data types and the data type definitions for the referenced data types created at compile time by resolving the referenced data types bottom up into data types known at runtime;
create a data object having the runtime data type based on the runtime data type definition; and
perform type checking on uses of the data object at runtime according to the runtime data type.
2. (Original) The product of claim 1, wherein:
instructions operable to create a data object having the runtime data type comprise instructions operable to assign the runtime data type to a data object.
3. (Currently Amended) The product of claim 1, wherein:
the data types included in the compound structure and used to create the runtime data type definition ~~types known at runtime~~ comprise data types defined by type definitions generated by compiling the source code program.

4. (Currently Amended) The product of claim 1, wherein:
the instructions to create a runtime data type and the instructions to use the runtime data
type are executed by a virtual machine; and

type definitions for the data types known at runtime the data type definitions for the
referenced data types are stored in a storage area managed by the virtual machine.

5. (Original) The product of claim 4, wherein:
the runtime type definition is stored in a local area for the computer program managed by
the virtual machine.

6. (Original) The product of claim 1, wherein:
the runtime type definition is a type object.

7. (Original) The product of claim 6, wherein:
the type object is managed as an element in a class hierarchy.

8. (Original) The product of claim 6, wherein:
the type object is subject to garbage collection.

9. (Original) The product of claim 6, wherein:
the type object is referenced programmatically in the source code program by a handle
and not by a name.

10. (Currently Amended) A computer system comprising:
one or more programmable processors and a machine-readable storage device;
a compile-time executable software module, stored on the machine-readable storage
device, including instructions that, when executed by the one or more programmable processors,
are operable to:

compile a procedural source code program to generate a compiled program
having instructions to create a runtime data type having a compound structure of
referenced data types and having instructions to use the runtime data type; and

create, at compile time, data type definitions for the referenced data types included in the compiled program and known at compile time; and
a runtime executable software module, stored on the machine-readable storage device, including instructions that, when executed by the one or more programmable processors, are operable to:

execute, at runtime, the instructions to create a runtime data type having a compound structure by creating a runtime data type definition from based on the compound structure of referenced data types and the data type definitions for the referenced data types created at compile time by resolving the referenced data types bottom-up into data types known at runtime;

create a data object having the runtime data type based on the runtime data type definition; and

perform type checking on uses of the data object at runtime according to the runtime data type.

11. (Original) The system of claim 1, wherein:
creating a data object having the runtime data type comprises assigning the runtime data type to a data object.

12. (Currently Amended) The system of claim 1, wherein:
the data types included in the compound structure and used to create the runtime data type definition ~~types known at runtime~~ comprise data types defined by type definitions generated by compiling the source code program.

13. (Currently Amended) The system of claim 1, wherein:
the runtime executable software module is a virtual machine.

14. (Currently Amended) The system of claim 1, wherein:
the ~~new type object~~ runtime type definition is a type object.

15. (Currently Amended) A method comprising:

compiling a procedural source code program to generate a compiled program having instructions to create a runtime data type having a compound structure of referenced data types and having instructions to use the runtime data type;

creating, at compile time, data type definitions for the referenced data types included in the compiled program and known at compile time;

executing, at runtime, the instructions to create a runtime data type having a compound structure by creating a new data type definition from based on the compound structure of referenced data types and the data type definitions for the referenced data types created at compile time ~~by resolving the referenced data types bottom-up into data types known at runtime;~~

creating a data object having the runtime data type based on the new data type definition;
and

performing type checking on uses of the data object at runtime according to the runtime data type.

16. (Original) The method of claim 15 wherein

creating a data object having the runtime data type comprises assigning the runtime data type to a data object.

17. (Currently Amended) The method of claim 15, wherein:

the types included in the compound structure and used to create the runtime data type definition ~~data types known at runtime~~ comprise data types defined by type definitions generated by compiling the source code program.

18. (Currently Amended) The method of claim 15, wherein:

the new data type definition ~~new type object~~ is a type object.

19. (New) The product of claim 1, wherein:

the runtime data type is defined by a programmer to create a new data type dynamically at runtime.

20. (New) The product of claim 1, wherein:
the data type definitions for the referenced data types included in the compiled program and known at compile time includes:
a first data type definition that defines a first structure that includes a first data component, and
a second data type definition that defines a second data component not included in the first structure, the second data type definition being different from the first data type definition and the second data component being different from the first data component;
the compound structure of referenced data types comprises a compound structure that includes at least the first data component included in the first structure and the second data component; and
creating a runtime data type definition based on the compound structure of referenced data types and the data type definitions for the referenced data types created at compile time comprises creating a runtime data type definition that defines a second structure that includes the first data component included in the first structure and the second data component.